updated "ghost" image of the rectangle (like a rubber band attached to the mouse), as well as the ability to change corners, and specify a minimum size. Previously, this was done by placing only the upper left and lower right corners, with no ghost box, and only a beep to indicate that the box was too small.

- BACKGROUNDS providing a "curtain" between windows representing active applications, and temporarily inactive ones. In the desk-top metaphor, this adds a drawer to the desk. A menu of background operations, as well as eye-pleasing images, are also provided.

- DEEXPOSED-MOUSE which allows windows to handle mouse clicks and documentation even when they are not completely exposed.

- SNAPSHOT-WINDOWS which allow the user to copy a portion of the screen, thus saving it for later use.

- TRANSPARENT-WINDOW which allows the image under a window to bleed through, providing the illusion of non-rectangular windows.

- An on screen round analog clock, with sweep second hand.

- Development tool consistency enhancements, including:

  - commands in the debugger, data structure inspectors (regular and flavor), editor so that they can call each other

  - commands in tools that do not have them to call the Lisp evaluator, obtain argument lists, obtain macro expansions, call the compiler, trace function invocation, and obtain programmer supplied documentation.

  - showing editor buffer reading status in a fashion similar to file reading status at the bottom of the screen

  - the ability to call the debugger on stack groups from many different contexts

  - the ability to modify entries in inspect panes in applications other than the Inspector

- Facilities to display a graph of time-varying quantities. This facility is useful for monitoring system performance parameters, such as the number of network packets sent or received, the number of disk operations per second, or the amount storage allocated.

- A Screen Saver that shuts the display video off after about twenty minutes of keyboard idleness to reduce display phosphor deterioration.

- A version of the terminal emulator program that does not take up the entire screen, and can have user configurable fonts.

- A facility for attaching functions to arbitrary keyboard keys, most commonly used to cause a particular instantiation of an application to be selected when a key is pressed, allowing rapid movement among applications.

- A number of editor commands, including Tags Compile Macro Calls, Macro Expand Into Buffer, Rotate Buffer, Rotate Buffer Backwards, Add File To Tag Table, Remove File From Tag Table, and Evaluate And Insert Into Buffer.

- Functions that allow the user to map over a set of files, applying a function to each file. The set of files can be specified very generally, and values accumulated in various ways from the mapped function.
- Extensions to the flavor inspector allowing it to function in many situations where it would previously have failed.
- A tools for displaying data organized hierarchically in trees, or as graphs, featuring
  - full cycle detection and handling,
  - mouse sensitive nodes and edges,
  - dynamic editing of the graph display,
  - horizontal and vertical scrolling,
  - an "overview" mode to facilitate moving the view port around in a large graph
- Introductory documents which have been used as models by a number of sites.

Of course, all of these will be provided to the user's library, and many of them have already been given to other sites, including Intellicorp, Berkeley, ISI, University of Maryland, and Ohio State.

In addition to producing and maintaining these software tools, we attempt to provide extensive testing and evaluation of Explorer hardware and software products in a sophisticated university research environment in order that these products work more effectively when they are distributed to the national community. This testing is critical to the development of the computing environment since the combination of concentrated in-house expertise and close links to the product developers allows a turnaround on problem fixes unavailable in the broader scope.

This year we have participated in testing TI's implementation of the Network File System protocol, Release 3.0 of the Explorer System Software, and Release 2.0 of TCP/IP. Our testing of NFS, besides uncovering the usual set of bugs, has allowed us to make suggestions to TI that have led to an order of magnitude increase in the data throughput of the implementation. Similarly, our experience with DARPA Internet protocols has allowed us to make many suggestions for improving the Release 3.0 Namespace System which TI has claimed to be invaluable in making the system acceptable to the many Arpanet users in the national community.

We served as a test site for several hardware revisions, as well, and further plan to perform extensive testing of the Explorer II VLSI-based machine when it becomes available in late spring or early summer.

Third party software is less utilized, but we stay abreast of the latest releases of the expert system shell KEE, and will be evaluating the Scribe text formatting system on the Explorer in a matter of weeks.

In addition to specific testing and evaluation, we are constantly finding, tracking, fixing, and reporting software bugs. This year we submitted thirty-two new bug reports on Release 2.1, twenty-one of which had fixes included. All of these fixes have been made available to the national community in a patch file.

There have been fifteen formal reports, with ten fixes on release 3 beta in the current four weeks of testing. There were forty-two reports returned with the TI representative who brought the initial software after the first week, most of which have been fixed by TI.

We have also worked extensively on the operational issues involved in keeping the machines running and useful from day to day. Texas Instruments had no hardware or software maintenance plans for large university installations in place, so we worked quite hard to engineer fair and serviceable plans for maintenance, resulting in the current offerings from TI for all university sites.

As well as working on these specific problems, we have had many meetings with Texas Instruments representatives wherein we have attempted to present the needs of the national community for short- and long-term AI workstation products, covering issues including the desirability of specialized hardware, address space, programming environment versus execution speed, and the ability to utilize the AI workstation's power for routine tasks.

Of course, there is also a large number of day-to-day activities needed to keep the computing environment pleasant, including resource management (e.g., disk space allocation, printer management), assistance with file backup and magnetic tape usage, and introducing new users to the system. We have produced documents targeted at complete novice users, users of InterLisp-D machines, and users of Symbolics machines in order to facilitate user education. These documents have been used as examples at various places in the national community.

For the coming year we plan to continue development and maintenance of the software tools, perhaps adding tools such as a DARPA Internet Domain Resolver, text processing facilities such as TeX, LaTeX, and document previewing tools, as well as aiding the growth of the users' group.

9.5 - Symbolics

Symbolics

Our work with Symbolics equipment has been slowed pending resolution of long-standing maintenance issues. As has been stated previously, in order for workstations to be competitive with time-shared mainframe computing resources, they must not only have a low purchase price, but must be cost-effective to maintain. This goal is normally achieved due to the economies of scale associated with having a large number of identical parts in an installation, as well as amortizing the cost of software development over many machines. We have come to reasonable agreements with all of the workstation vendors except for Symbolics. The high costs of service, the exceptionally high price of mail-in board repair, and the lack of a reasonable self-service alternative has left us unable to justify continued support of these machines unless a workable agreement can be reached. We have negotiated a tentative hardware maintenance contract, involving parts from Symbolics, and in-house labor, and are in negotiations for software maintenance. If we can reach consensus, we will be able to increase support of Symbolics machines once again.

While there have been no appreciable system development activities with the Symbolics machines, they have been maintained in good working order, with up-to-date software. We have not, however, moved from Release 6.1 to Genera 7.0 as the user community felt that the disadvantages of the transition overwhelmed the advantages, since the new software was quite incompatible with existing code, was slower, and seemed to introduce many new problems. We will re-evaluate Genera 7.1 when it is released. KEE and Fortran have been kept current, and patches in bulletins from Symbolics have been applied.

9.6 - SUN

We are just now bringing up several SUN workstations configured for Lisp research work. Several SUMEX projects have been able to experiment with SUN workstations through collaborations with other groups and the Lisp programming and debugging environments of these machines is still rather primitive as compared to the InterLisp and ZetaLisp machine environments. Also, SUN's need to be configured with relatively large memories to accommodate Lisp systems (because of limited garbage collection facilities currently) and this has required using third part memory. More standard configurations should be available from vendors shortly and we expect to have additional information to report next year.

10 - Workstation Standards and Access

10.1 - Computing Environment Standards

In a heterogeneous computing environment, such as AI research inevitably involves, the issue of cross-system compatibility is a central one. Users of various machines want to be able to share software, as well as be able to use various machines with a minimum of overhead in learning the operating procedures and programming languages of new systems. Thus, it is crucial to specify and propagate powerful, flexible standards for various aspects of the computing environment so that it is possible to transfer both skills and information among machines.

In order to improve the inter-machine compatibility of our software, we have been encouraging all users to use the CommonLisp programming language [16], as well as pressing vendors to provide more complete and efficient implementations of this language. We have already served as beta test sites for Xerox, Texas Instruments, and Lucid CommonLisp implementations.

The CommonLisp language, however, is only a subset of the software needed for our research. Research projects need higher-level powerful facilities, such as an object-oriented programming system and sophisticated error handling. Therefore we have been supporting and following the development of the CommonLisp Object System (CLOS) via membership in the electronic discussion group, technical contributions, and porting of Portable Common Loops (PCL), a predecessor of CLOS, to the TI Explorer. We are now encouraging vendors to produce efficient implementations of the system, and users to familiarize themselves with it. We are also encouraging vendors to adopt the proposed CommonLisp error system.

Other features of the computing environment also need to be standardized to be useful on more than one machine at a time. Another of the most important of these is the keyboard and display interface, often referred to as the "window system". See the virtual graphics section (page 34) for further discussion of window systems.

There are also many other areas which could benefit greatly from standardization, including document page description languages, text and graphics representations, and more networking protocols. However, it is important that standards not be entered into hastily, as an insufficient standard can often be worse than no standard at all. We intend to continue working to develop standards for these and other computing needs as the understanding of the issues involved matures.

10.2 - Protocol Standards

In addition to various portions of the AI research computing environment, the most highly visible area of standardization has been inter-machine communication, or networking. Underlying all network I/O must be a network protocol for packet transfer between cooperating hosts. At SUMEX we have had long term experience with several such protocols; PUP/BSP, PUP/EFTP, IP/TCP, IP/TFTP, IP/UDP, IP/SMTP, and NS/SPP are those most commonly used on SUNet. PUP/BSP and IP/TCP have been used to implement both FTP and TELNET, PUP/EFTP is an "Easy File Transfer Protocol" on top of PUP used for boot like services. IP/TFTP is a "Trivial File Transfer Protocol" which uses IP/UDP datagrams. IP/SMTP is the "Simple Mail Transfer Protocol" for sending mail, and runs on top of IP/TCP. NS/SPP is a "Sequenced Packet Protocol" similar to PUP/BSP and is used for FTP and TELNET. In the past we have elected to write servers for each new protocol in order to accommodate both vendor hardware and systems software requirements. This was necessary because no one protocol has been supported on all such systems.

With others in the computer science research community, we have pressed vendors to supply implementations of the DARPA standard TCP/IP communications protocols. We are pleased that the IP protocol family is now supported on all hardware and operating system configurations currently at SUMEX. And we expect to have IP support on any new systems we purchase in the future. Similarly, IP is supported on all of our UNIX based file servers, and the SUNet gateways route all IP datagrams. There has been a great deal of deliberate effort at Stanford and SUMEX to enforce IP as a standard protocol for new software development. This was motivated by its broad acceptance and the growing number implementations throughout the networking and vendor communities. This does not imply that we will abandon the other protocols but rather, since we are seeking to have *uniformity across all vendors* with the proposed Stanford distributed environment, we are choosing to limit new implementations to the IP protocol family. We are also currently working to provide improved support for TCP/IP in our Terminal Interface Processors (TIP's), having already implemented TCP/IP routing service.

As an example of the power of using uniform communication protocols, we set up a Xerox 1186 workstation for use by Dr. Shortliffe during his sabbatical in Philadelphia at the University of Pennsylvania. This university has a different network environment than Stanford's, although it is probably more typical of common Ethernet installations. The Pennsylvania network provides only Class-B IP/TCP services for VAX-based VMS/Unix systems over "thin" Ethernet. The 1186 was the only piece of Xerox hardware on the network so the disk was pre-loaded at Stanford. We successfully used the Pennsylvania VMS VAX's as file servers and time servers (after writing appropriate software to interface the workstation to the RFC868 time protocol). Using their Ethernet to ARPANET gateway, we were able to connect to SUMEX-AIM directly from the Xerox workstation as well as access our print servers at Stanford. Unfortunately, hardware problems with the workstation later in the year prevented us from attempting any more complex experiments with distributed computing and remote hosts.

Such standardization has a price, however, in that observed network communications speeds are uniformly higher between equipment "tuned" to individual vendor protocols. For a discussion of network file access protocol benchmarks see page 40.

11 - Network Services

A highly important aspect of the SUMEX system is effective communication within our growing distributed computing environment and with remote users. In addition to the economic arguments for terminal access, networking offers other advantages for shared computing. These include improved inter-user communications, more effective software sharing, uniform user access to multiple machines and special purpose resources, convenient file transfers, more effective backup, and co-processing between remote machines. Networks are crucial for maintaining the collaborative scientific and software contacts within the SUMEX-AIM community.

11.1 - Remote Networks

11.1.1 - Commercial Network Link

At the beginning of this grant year, SUMEX had just begun switching public data networks (PDN) vendors (from TYMNET to UNINET) in an attempt to improve service for our users. As the result of a corporate merger, our connection to UNINET became a connection to TELENET.

11.1.2 - X.25/Ethernet Link

SUMEX and Stanford's heavy use of Ethernet has prompted interest in a suitable connection between our Ethernet system and the Public Data Networks (specifically TELENET). Commercial groups provide a wide variety of equipment connecting these X.25 networks to Ethernets, but lack of standards for terminology make it difficult to determine their function.

Because our interest involves connection to other X.25 based hosts and Packet Assemblers/Disassemblers (PADs), sometimes called Terminal Interface Processors, as opposed to connecting two Ethernets via an X.25 net, we need a device that provides protocol translation. One alternative we are considering is to use a SUN processor for this task. This processor will have its normal TCP/IP Ethernet capability supplemented by an X.25 package provided by SUN. Such a package will provide SUMEX with both an inbound and outbound capability relative to TELENET. Users on SUMEX will be able to access the large variety of hosts and services on the PDNs (such as NLM and Dialog) in a simple and reliable manner. Though the high level protocols for file transfer and mail exchange are developing slowly in the X.25 environment, some progress is being made, so a general purpose interface to these networks is an important asset.

11.1.3 - ARPANET Link

We also continue our extremely advantageous connection to the Department of Defense's ARPANET, managed by the Defense Communications Agency (DCA). This connection has been possible because of the long-standing basic research effort in AI within the Knowledge Systems Laboratory that is funded by DARPA. ARPANET is the primary link between SUMEX and other university and AIM machine resources, including the large AI computer science community supported by DARPA. We are also attempting to establish a link to the DARPA wideband satellite network to facilitate the rapid transfer of large amounts of data such as are involved with projects like our Concurrent Symbolic Computing Architectures project.

As a member of the ARPAnet group, we have an obligation to help with certain network operations tasks. For instance, we participated in the upgrading (to 56 Kbs) of the connection which Advanced Decision Systems, Inc. (Mt. View, CA) has to our IMP. We also have a minor role in certain mail routing functions for the ARPANET community.

As part of an overall increase in ARPANET capacity a third 56 Kbs trunk line is being added to our IMP by the Defense Communications Agency (DCA).

11.2 - Microcomputer Networks

We connected our Apple Macintosh computers in 2 buildings with *Appletalk* and *Phonenet* network products. More significantly, we integrated them with the rest of our equipment by connecting the microcomputer networks to the campus Ethernet networks using *Kinetics FastPath* gateways, a commercial spinoff resulting from the SUMEX work on the SEAGATE gateway.

Software written at Columbia University, Stanford, and elsewhere, makes it possible for a Macintosh to share a VAX file server with the Lisp machines and to access hosts on the ARPA internet as a first-class workstation.

Usage of a centralized VAX file server makes nightly backup and data sharing automatic. This mode of usage is a great improvement over the isolated stand-alone machines that most people think of when they think of microcomputers.

11.3 - Local Area Networks

For many years now, we have been developing our local area networking systems to enhance the facilities available to researchers. Much of this work has centered on the effective integration of distributed computing resources in the form of mainframes, workstations, and servers. Network gateways and terminal interface processors (TIP's) were developed and extended to link our environment together. We are developing gateways to interface other equipment as needed too. A diagram of our local area network system is shown in Figure 8 and the following summarizes our LAN-related development work.

11.3.1 - Ethernet Gateways

In our heterogeneous network environment, in order to provide workstation access to file servers, mail servers, and other computers within the network, it is necessary to able to route multiple networking protocols through the network gateways. As reported last year, the SUMEX gateways support PUP, Xerox NS, Symbolics/Texas-Instrument CHAOSNET, and the IP/TCP protocols. This support not only provides the routers necessary to move such packets among the subnetworks, but also other miscellaneous services such as time, name/address lookup, host statistics, boot strap support, address resolution, and routing table broadcast and query information.

This year, with the acquisition of a SMI SUN 3/180 file server and three SUN 3/75 workstations, it was necessary to add special boot-protocol support for SMI's Net Disk and NFS protocols to allow the SUN workstations to boot their Unix kernel, and runnable programs while residing on a network that is distinct from the one on which the file server resides. SUN's convention is that each subnet must have its own file server that can provide boot support. But this is too expensive for complex network environments such as ours. Given this broadened capability, we can now place our "diskless" SUN workstations anywhere within the KSL network topology, rather than on the same network that the server resides.

Also, to improve the throughput of our highly loaded gateways, portions of Ethernet interface drivers and protocol routers were rewritten. The drivers now look for the arrival of additional packets while processing those packets that initiated the interrupt. Now, each router can process up to six packets before relinquishing control to the gateways process scheduler. Previously, each router would process only one such packet per call. These two changes more than doubled the maximum observed packets per second, as well as the maximum throughput bandwidth which is now about 2.5 megabits per second, and minimized the dropping of back-to-back packets by the Ethernet interface itself.

Over the past year our network topology grew in complexity and extent so that we now have redundant routes to several networks within the KSL and Stanford LAN. Within this more complex environment, the old routing table management schemes broke down and had to be redesigned and changed to adequately deal with the network interactions that arose. In particular, we had to ensure that when a route to a particular network no longer was available because of electrical, hardware or software failure, that this information was propagated throughout the topology in a manner that maintained routing table equilibrium. We have solved this problem and our gateways now recover gracefully from these failures.

A second kind of failure occurs when a path between two networks fails but the gateways involved are not aware of this fact, and as a consequence continue to advertise routes using paths that are partitioned. We have had two examples of this over the past year caused by the failure of a repeater in one case and a transceiver in the other. When we detect such a situation, we can now remove the route from the gateway generating it using software, make the repair, and then replace the route, without perturbing the connectivity of our topology if there are redundant routes around the partition caused by hardware failure.

Finally, a minor change in the gateways' routing table update algorithm when multiple routes to a network are available has managed to balance the load between these alternative paths, and increase the throughput at the gateways involved. Such gateways are usually focal points for high traffic volume, and the change was immediately noted by staff members sensitive to network throughput. The old version of the routing update protocol would hold onto a route even if alternative paths of equal cost were available. The new version will always update a route if a path of equal cost arises. When n redundant paths are available, the route changes approximately every 30/n seconds.

These services are still unique within the SUMEX-AIM portion of the Stanford University network, and give our researchers a networking environment that is flexible, of high bandwidth, and extremely dependable.

11.3.2 - Terminal Interface Processors

With the advent of reliable multiple speed (300, 1200 and 2400 baud) modems, we placed ten such devices on our TIPs for dial-in access, and added autobaud recognition to the TIP software. 2400 baud dial-in connections have shown themselves to be highly responsive in such a configuration, and have the advantage when placed on the TIP of giving the user access to any host on the Stanford local area network. Autobaud recognition has also been added to the directly attached tty ports to simplify user/TIP interaction. If a user changes his terminal's baud rate, the TIP will still be responsive, but at a different speed. Previously, such a line's baud rate was fixed, and this often led to a great deal of user frustration.

Also, the experimental NTT ELIS Lisp Machines used in the KSL currently do not have Ethernet connections. To accommodate remote access to these systems, they were attached to TIP ports so that a user could connect to the TIP from the Ethernet, and then transparently connect to the ELIS machines via a TIP command. Once this connection is established, the user appears to have a terminal directly attached to the ELIS itself. Currently, there are six such ELIS ports in use on one of our TIPs. Incidentally, the same code is currently being generalized for use as a dial-out module.

12 - Printing Services

Laser printers have become essential components of the work environment of the SUMEX-AIM community with applications ranging from scientific publications to hardcopy graphics output for ONCOCIN chemotherapy protocol patient charts. We have done much systems work to integrate laser printers into the SUMEX network environment so they would be routinely accessible from hosts and workstations alike. This software has been widely shared with other user groups in the AIM community and beyond.

SUMEX operates 7 medium-speed (8-20 pages per minute) Imagen laser printers, 2 low-speed (~3 ppm) Xerox laser printers, and 1 low-speed (~3 ppm) Apple laser printer. Each of the Imagen printers possesses an emulator for a line printer, a daisy wheel printer, a Tektronix plotter, and a typesetter (using the *Impress* language). The last 3 printers render the special-purpose *Press*, *Interpress*, and *Postscript* typesetter languages. In total, the laser printers printed about half a million pages of output during the year. Most of the printout was simple text, followed in quantity by formatted text in *Impress* format, Impress-format drawings, and screen dumps. Lastly, about 2000 pages each of Postscript-format drawings and formatted text were printed on the Apple Laser Writer. Although the *Postscript* language is probably the most popular typesetting language among commercial applications developers at the present time (and one which we support with the Laser Writer), the overwhelming preponderance of readily-renderable line printer and Impress jobs in our printing mix provides the basis for our decision to emphasize the relatively high-speed Imagen laser printers. Because of the increasing usage of Postscript among vendors, however, we have purchased an additional Apple Laser Writer for use in the Medical School Office Building.

In order to finally obtain families of fonts in common between our Press, Impress and Interpress printers, we used the TypeFounder software that we beta-tested for Xerox to extract font width information (for use by our workstations) from our existing Interpress printer fonts (a 12 page per minute, 300 dpi printer based on the Xerox 8000 processor) and also made new fonts using character splines from an earlier Xerox grant program. Having an overlap in fonts among the printers helps to relieve the problems inherent in trying to print the same complex document on different printer technologies. Some of the font additions required software patches for the Interpress driver software on the workstations. The Interpress driver was further modified to provide rotated fonts in order to print our specialized medical forms.

13 - General User Software

We have continued to assemble (develop where necessary) and maintain a broad range of user support software. These include such tools as language systems, statistics packages, vendor-supplied programs, text editors, text search programs, file space management programs, graphics support, a batch program execution monitor, text formatting and justification assistance, magnetic tape conversion aids, and user information/help assistance programs.

A particularly important area of user software for our community effort is a set of tools for inter-user communications. We have built up a group of programs to facilitate many aspects of communications including interpersonal electronic mail, a "bulletin board" system for various special interest groups to bridge the gap between private mail and formal system documents, and tools for terminal connections and file transfers between SUMEX and various external hosts. Examples of work on these sorts of programs have already been mentioned in earlier sections, particularly as they relate to extensions for a distributed computing environment.

At SUMEX-AIM we are committed to importing rather than reinventing software where possible. As noted above, a number of the packages we have brought up are from outside groups. Many avenues exist for sharing between the system staff, various user projects, other facilities, and vendors. The availability of fast and convenient communication facilities coupling communities of computer facilities has made possible effective intergroup cooperation and decentralized maintenance of software packages. The many operating system and system software interest groups (e.g., TOPS-20, UNIX, D-Machines, network protocols, etc.) that have grown up by means of the ARPANET have been a good model for this kind of exchange. The other major advantage is that as a by-product of the constant communication about particular software, personal connections between staff members of the various sites develop. These connections serve to pass general information about software tools and to encourage the exchange of ideas among the sites and even vendors as appropriate to our research mission. We continue to import significant amounts of system software from other ARPANET sites, reciprocating with our own local developments. Interactions have included mutual backup support, experience with various hardware configurations, experience with new types of computers and operating systems, designs for local networks, operating system enhancements, utility or language software, and user project collaborations. We have assisted groups that have interacted with SUMEX user projects get access to software available in our community (for more details, see the section on Dissemination on page 103).

III.A.3.5. Relevant Core Research Publications

The following is a list of new publications and reports that have come out of our core research and development efforts over the past year:

KSL 85-57
(Journal Memo) E. Horvitz and D. Heckerman; **The Inconsistent Use of Measures of Certainty in Artificial Intelligence Research,** August 1985. To appear in: *Uncertainty in Artificial Intelligence* **15 pages**

KSL 85-58
(Journal Memo) C.D. Lane, M.E. Frisse, L.M. Fagan, and E.H. Shortliffe; **Object-Oriented Graphics in Medical Interface Design,** December 1985. To appear in: *AAMSI-86* **5 pages**

KSL 85-59
(Working Paper) Allan Terry; **Using Explicit Strategic Knowledge to Control Expert Systems,** December 1985. Submitted for publication in: *Artificial Intelligence* **51 pages**

KSL 85-60
(Working Paper) Jean-Luc Bonnetain; **FLOWER: A First Cut at Designing a Budget Proposal,** September 1985. **28 pages**

KSL 86-18
STAN-CS-86-1123. H. Penny Nii; **Blackboard Systems,** June 1986. To appear in: *AI Magazine Vols. 7-2 and 7-3.* **86 pages**

KSL 86-24
(Journal Memo) M.A. Musen, L.M. Fagan, D.M. Combs, and E.H. Shortliffe; **Using a Domain Model to Drive An Interactive Knowledge Editing Tool,** September 1986. To appear in: *Proceedings of AAAI Workshop on Knowledge Acquisition, 1986* **12 pages**

KSL 86-25
(Journal Memo) E.J. Horvitz, D.E. Heckerman, and C.P. Langlotz; **A Framework for Comparing Alternative Formalisms for Plausible Reasoning,** May 1986. **5 pages**

KSL 86-28
(Working Paper) James Brinkley, Craig Cornelius, Russ Altman, Barbara Hayes-Roth, Olivier Lichtarge, Bruce Duncan, Bruce Buchanan, Oleg Jardetzky; **Application of Constraint Satisfaction Techniques to the Determination of Protein Tertiary Structure,** March 1986. **14 pages**

KSL 86-29
(Working Paper) Matthew L. Ginsberg; **Multi-valued logics,** April 1986. To appear in: *AAAI - 86* **13 pages**

KSL 86-33
(Journal Memo) David E. Heckerman and Eric J. Horvitz; **The Myth of Modularity in Rule-Based Systems,** May 1986. **7 pages**

KSL 86-36
STAN-CS-87-1148. Bruce A. Delagi, Nakul Saraiya, Sayuri Nishimura, and Greg Byrd; **An Instrumented Architectural Simulation System,** January 1987. **21 pages**

KSL 86-37
(Working Paper) Matthew L. Ginsberg; **Possible Worlds Planning,** April 1986. Submitted for publication to: *1986 Planning Workshop* **13 pages**

KSL 86-38
STAN-CS-87-1147. Barbara Hayes-Roth, M. Vaughan Johnson Jr., Alan Garvey, and Michael Hewett; **A Modular and Layered Environment for Reasoning about Action,** April 1987. To appear in: *The Journal of Artificial Intelligence in Engineering, Special Issue on Blackboard Systems, October 1986.* **63 pages**

KSL 86-39
(Journal Memo) E.H. Shortliffe; **Artificial Intelligence in Management Decisions: ONCOCIN,** April 1986. To appear in: *Proceedings of a Conference on Medical Information Sciences, University of Texas Health Sciences Center at San Antonio, July 1985. Also in Frontiers of Medical Information Sciences, Praeger Publishing, 1986.* **14 pages**

KSL 86-40
(Journal Memo) Christopher Lane; **The Ozone Manual,** July 1986. **34 pages**

KSL 86-42
(Working Paper) Oleg Jardetzky, Andrew Lane, Jean-Francois Lefevre, Olivier Lichtarge, Barbara Hayes-Roth, Russ Altman, Bruce Buchanan; **A New Method for the Determination of Protein Structures in Solution from NMR,** May 1986. Submitted for publication in: *Proc. XXIII Congress Ampere, Rome, Italy, Sept. 1986* **6 pages**

KSL 86-43
(Journal Memo) Edward H. Shortliffe; **Update on Oncocin: A Chemotherapy Advisor for Clincal Oncology,** August 1986. Submitted for publication in: *Medical Informatics* **4 pages**

KSL 86-44
(Thesis) Stephen M. Downs; **A Program for Automated Summarization of On-Line Medical Records,** June 1986. **27 pages**

KSL 86-46
STAN-CS-86-1111. Paul Rosenbloom and John Laird; **Mapping Explanation-Based Generalization onto Soar,** June 1986. To appear in: *AAAI-86* **18 pages**

KSL 86-47
STAN-CS-86-1124. Daniel J. Scales; **Efficient Matching Algorithms for the SOAR/OPS5 Production System,** June 1986. **50 pages**

KSL 86-48
(Working Paper) William J. Clancey; **Review of Winograd and Flores' "Understanding Computers and Cognition: A New Foundation for Design",** July 1986. **13 pages**

KSL 86-49
(Journal Memo) M.A. Musen, D.M. Combs, J.D. Walton, E.H. Shortliffe, L.M. Fagan; **OPAL: Toward the Computer-Aided Design of Oncology Advice Systems,** July 1986. Submitted for publication to: *Proceedings of the Tenth Annual Symposium on Computer Applications in Medical Care.* **10 pages**

KSL 86-50
(Working Paper) Ross D. Shacter and David E. Heckerman; **A Backwards View for Assessment,** July 1986. **6 pages**

KSL 86-51
Barbara Hayes-Roth, Bruce Buchanan, Olivier Lichtarge, Michael Hewett, Russ Altman, James Brinkley, Craig Cornelius, Bruce Duncan, and Oleg Jardetzky; **PROTEAN: Deriving protein structure from constraints,** March 1986. To appear in: *Proceedings of AAAI 1986* **21 pages**

KSL 86-52
(Working Paper) Edward H. Shortliffe, M.D., Ph.D; **Medical Expert Systems: Knowledge Tools for Physicians**, September 1986. **24 pages**

KSL 86-53
(Working Paper) Edward H. Shortliffe, M.D., Ph.D; **Medical Expert Systems Research at Stanford University**, September 1986. **13 pages**

KSL 86-56
(Working Paper) Nakul P. Saraiya; **AIDE: A Distributed Environment for Design and Simulation** June 1986. **25 pages**

KSL 86-57
(Working Paper) Curtis P. Langlotz, Edward H. Shortliffe, and Lawrence M. Fagan; **A Methodology for Computer-Based Explanation of Decision Analysis**, November 1986. **21 pages**

KSL 86-58
William J. Clancey; **Intelligent Tutoring Systems: A Tutorial Survey**, September 1986. Submitted for publication in: *Collected papers of the International Professorship in Computer Science (Expert Systems) Universite de L'Etat, Belgium* 43 pages

KSL 86-60
(Working Paper) Alan Garvey, Michael Hewett, M. Vaughan Johnson, Robert Schulman, Barbara Hayes-Roth; **BB1 User Manual - Interlisp Version**, October 1986. **68 pages**

KSL 86-61
(Working Paper) Alan Garvey, Michael Hewett, M. Vaughan Johnson, Robert Schulman, Barbara Hayes-Roth; **BB1 User Manual - Common Lisp Version**, October 1986. **72 pages**

KSL 86-62
(Working Paper) David C. Wilkins; **On the Limits of Debugging via Differential Modeling**, October 1986. **15 pages**

KSL 86-63
(Working Paper) David C. Wilkins; **Knowledge Base Debugging Using Apprenticeship Learning Techniques**, October 1986. **15 pages**

KSL 86-64
(Working Paper) Donald E. Henager; **Window-Driven Object-Oriented Calculator**, March 1986. **56 pages**

KSL 86-65
Matthew L. Ginsberg, David E. Smith; **Reasoning About Action I: A Possible Worlds Approach**, May 1987. **25 pages**

KSL 86-66
Matthew L. Ginsberg, David E. Smith; **Reasoning About Action II: The Qualification Problem**, May 1987. **28 pages**

KSL 86-68
David E. Smith; **Controlling Backward Inference**, March 1987. **67 pages**

KSL 86-69
STAN-CS-86-1136. Harold Brown, Eric Schoen, and Bruce Delagi; **An Experiment in Knowledge-Base Signal Understanding Using Parallel Architectures**, October 1986. To appear in: *Parallel Computation and Computers for AI, J.S. Kowalik Editor, Kluwer Publishers.* 39 pages

KSL 86-70
STAN-CS-86-1140. John E. Laird, Allen Newell, and Paul S. Rosenbloom; **Soar: An Architecture for General Intelligence,** December 1986. To appear in: *Artificial Intelligence*. **66 pages**

KSL 86-74
(Thesis) Glenn Douglas Rennels; **A Computational Model of Reasoning from the Clinical Literature,** June 1986. **244 pages**

KSL 86-75
(Journal Memo) Eric J. Horvitz; **Toward a Science of Expert** Systems, March 1986. **8 pages**

KSL 86-76
M. Vaughan Johnson Jr. and Barbara Hayes-Roth; **Integrating Diverse Reasoning Methods in the BB1 Blackboard Control Architecture,** December 1986. **17 pages**

KSL 87-01
(Working Paper) David C. Wilkins, William J. Clancey, and Bruce G. Buchanan; **Knowledge Base Refinement Using Abstract Control Knowledge,** January 1987. **9 pages**

KSL 87-02
STAN-CS-87-1146. Gregory T. Byrd, Russell Nakano, and Bruce A. Delagi; **A Point-to-Point Multicast Communications Protocol,** January 1987. **30 pages**

KSL 87-03
Bruce G. Buchanan; **Artificial Intelligence** As An **Experimental** Science, January 1987. To appear in: *Synthese* **41 pages**

KSL 87-05
STAN-CS-87-1142. James F. Brinkley, Bruce G. Buchanan, Russ B. Altman, Bruce S. Duncan, Craig W. Cornelius; **A Heuristic Refinement Method for Spatial Constraint Satisfaction Problems,** January 1987. **15 pages**

KSL 87-06
(Journal Memo) Glenn D. Rennels; A **Computational Model of Reasoning from the Clinical Literature,** January 1987. To appear in: *SCAMC Proceedings, Washington D.C. 1986* **8 pages**

KSL 87-07
STAN-CS-87-1144. Gregory T. Byrd and Bruce A. Delagi; **Considerations for Multiprocessor Topologies,** January 1987. **6 pages**

KSL 87-08
Robert Schulman and Barbara Hayes-Roth; **ExAct: A Module for Explaining Actions,** January 1987. **15 pages**

KSL 87-09
(Working Paper) Peter D. Karp and Peter Friedland; **Coordinating the Use of Qualitative and Quantitative Knowledge in Declarative Device Modeling,** January 1987. **20 pages**

KSL 87-11
Alan Garvey, Craig Cornelius, and Barbara Hayes-Roth; **Computational Costs versus Benefits of Control Reasoning,** February 1987. **13 pages**

KSL 87-12
(Working Paper) William J. Clancey; **The Knowledge Engineer as Student: Metacognitive bases for asking good questions,** January 1987. To appear in *Learning Issues for Intelligent Tutoring Systems, Heinz Mandl and Alan Lesgold, editors. Springer-Verlag: New York* **30 pages**

KSL 87-16
(Journal Memo) Eric J. Horvitz; **Inference under Varying Resource Limitations,** February 1987. **16 pages**

KSL 87-18
Isabelle de Zegher-Geets, Andy Freeman, Mike Walker, Bob Blum, Gio Wiederhold; **Computer-Aided Summarization of a Time-Oriented Medical Database,** February 1987. **6 pages**

KSL 87-19
(Working Paper) Curt P. Langlotz and Edward H. Shortliffe; **The Relationship between Decision Theory and Default Reasoning,** February 1987. **16 pages**

KSL 87-20
(Working Paper) Michael G. Kahn; **Model-Based Interpretation of Time-Ordered Data,** March 1987. **18 pages**

KSL 87-21
(Working Paper) Gregory F. Cooper; **An Algorithm for Computing Probabilistic Propositions,** March 1987. **5 pages**

KSL 87-22
(Journal Memo) Homer L. Chin and Gregory F. Cooper; **Stochastic Simulation of Casual Bayesian Models,** March 1987. **11 pages**

KSL 87-23
(Working Paper) Thierry Barsalou and Gio Wiederhold; **Applying a Semantic Model to an Immunology Database,** March 1987. **18 pages**

KSL 87-24
(Journal Memo) Homer L. Chin and Gregory F. Cooper; **Knowledge-Based Patient Simulation,** March 1987. **11 pages**

KSL 87-25
(Journal Memo) Edward H. Shortliffe; **Computers in Support of Clinical Decision Making,** March 1987. **12 pages**

KSL 87-32
(Working Paper) William J. Clancey; **Diagnosis, Teaching, and Learning: An Overview of GUIDON2 Research,** April 1987. 12 Pages

KSL 87-34
(Working Paper) Russell Nakano; **Experiments with a Knowledge-Based System on a Multiprocessor: Preliminary AIRTRAC-LAMINA Qualitative Results.** June, 1987.

KSL-87-35
(Working Paper) Masafumi Minami; [Experiments with a Knowledge-Based System on a Multiprocessor: Preliminary AIRTRAC-LAMINA Quantitative Results.] June, 1987.

Other Outside Articles:

Hayes-Roth, B., Johnson, M.V., Garvey, A., and Hewett, M. Applications of BB1 to arrangement-assembly tasks. *Artificial Intelligence in Engineering*, October, 1986.

Hayes-Roth, B., Johnson, M.V., Garvey, A., and Hewett, M. The BB* environment. To appear in: R. Engelmore and A. Morgan (Eds.), *Blackboard Systems*. Addison-Wesley, London: 1987.

Garvey, A., Cornelius, C., and Hayes-Roth, B. Computational costs versus benefits of control reasoning. *Proceedings of the American Association for Artificial Intelligence: AAAI87*, 1987.

M. V. Johnson, and Hayes-Roth, B. Integrating diverse reasoning methods in BB1. *Proceedings of the American Association for Artificial Intelligence: AAAI87*, 1987.

Hayes-Roth, B. Blackboard systems. In Eckroth, D. (Ed.), *Encyclopedia of Artificial Intelligence*. New York: John Wiley & Sons, 1987.

Garvey, A., Hewett, M., Johnson, M.V., Schulman, R., and Hayes-Roth, B. *BB1 User's Manual*. Stanford University. 1987.

Hayes-Roth, B., Buchanan, B.G., Lichtarge, O., Hewett, M., Altman, R., Brinkley, J., Cornelius, C., Duncan, B., and Jardetzky, O. Elucidating protein structure from constraints in PROTEAN. To be reprinted in: R. Engelmore and A. Morgan (Eds.), *Blackboard Systems*. Addison-Wesley, London: 1987.

Duncan, B., Buchanan, B. G., Hayes-Roth, B., Lichtarge, O., Altman, R., Brinkley, J., Hewett, M., Cornelius, C., and Jardetzky, O. PROTEAN: A new method of deriving solution structures of proteins. *Bulletin of Magnetic Resonance*, 1987.

Hayes-Roth, B., Buchanan, B.G., Lichtarge, O., Hewett, M., Altman, R., Brinkley, J., Cornelius, C., Duncan, B., and Jardetzky, O. PROTEAN: Deriving protein structure from constraints. *Proceedings of the American Association for Artificial Intelligence: AAAI86*, 1986.

Jardetzky, O., Lane, A., Lefevre, J., Lichtarge, O., Hayes-Roth, B., Altman, R., and Buchanan, B. A new method for the determination of protein structures in solution from NMR. In *Proceedings of the XXIII Congress Ampere*, Rome, Italy, 1986.

Tommelein, I. D., Johnson, M. V., Hayes-Roth, B., and Levitt, R.E., SIGHTPLAN: A Blackboard Expert System for the Layout of Temporary Facilities on a Construction Site. *Proceedings of the IFIP WG5.2 Conference on Expert Systems in Computer-Aided Design*, Sydney, Australia, 1987.

Tommelein, I. D., Levitt, R. E., and Hayes-Roth, B., Using Expert Systems for the Layout of Temporary Facilities on Construction Sites." *Proceedings of the CIB W-65 Symposium on Organization and Management of Construction*, Birkshire, U.K, 1987.

III.A.3.6. Resource Equipment

The SUMEX-AIM core facility, started in March 1974, was built around a Digital Equipment Corporation (DEC) KI-10 computer and the TENEX operating system and continued through the 1970's with a mainframe focus for the resource. The interactive computing environment of this facility, with its AI program development tools and its network and interpersonal communication media, was unsurpassed in other machine environments. Biomedical scientists found SUMEX easy to use in exploring applications of developing artificial intelligence programs for their own work and in stimulating more effective scientific exchanges with colleagues across the country. Coupled through wide-reaching network facilities, these tools provided us access to a large computer science research community, including active artificial intelligence and system development research groups.

In the late 1970's and early 1980's, computer system research on early microprocessors and compact minicomputers suggested that large mainframe computers would not be essential or even the dominant source of computing power for AI research and AI program dissemination. Thus, we began to implement a strategy for computing resources marked by the integration of heterogeneous systems -- mainframes, Lisp workstations, and service systems (e.g., for file storage and printing) all linked together by local area networks. Over the years, we have configured the optimal resource computing environment around shared central machines coupled through a high-performance network to growing clusters of personal workstations.

The concept of the individual workstation, especially with the high-bandwidth graphics interface, proved ideal. Both program development tools and facilities for expert system user interactions were substantially improved over what is possible with a central time-shared system. The main shortcomings of early workstation systems were their limited processing speed and high cost. But in the few years since our first experimental systems, processing power has increased by more than a factor of 10 and the cost has decreased by a comparable factor.

Today the SUMEX resource is a complex, integrated facility comprised of machines, networks, and servers illustrated in Figures 4 - 8. A key role of the SUMEX-AIM resource is to continue to evaluate workstations as the technology is changing rapidly. This evaluation includes new hardware and software, 1) to provide superior development and execution platforms for AI research, and 2) to support the ancillary "office environment" (presently carried out on the DEC 2060, which is being phased out). Thus far no single workstation has materialized that provides all the services we would like to see in support of either or both of these missions. This means that for the foreseeable future, we will utilize a multiplicity of machines and software to address the needs of the projects.

Systems based on the Motorola 68020 chip (e.g., SUN Microsystems or Apple Macintosh II workstations), the Intel 80286 and 80387 chips (e.g., IBM PS/1-4 machines), and other newer architectures, such as reduced instruction set computer (RISC) chips, have Lisp benchmark data rivaling the performance of existing, specially microcoded Lisp machines (e.g., those from Xerox, Symbolics, and TI). But these Lisp machine vendors are producing substantially faster machines as well, using VLSI technology. It is still too early to predict how this "race" will ultimately turn out and software environments will play an equally important role to raw hardware speed in the decision. For now, the Lisp software environments on the "stock" machines are not nearly so extensively developed as on Lisp machines and conversely, the routine computing environments of Lisp machines (text processing, mail, spreadsheets, etc.) lag the tools available on stock UNIX machines.

In the past year we experimentally tried increasing usage of TI and Xerox Lisp machines (purchased as AI research platforms) for text editing and document formatting, but their functionality and speed do not approach that of the *TeX* and *Scribe* formatters when executed on the 2060. The Lisp machines do not yet provide the myriad tools of the 2060 (e.g. mail, database, spreadsheet, dictionary), but we and other groups are undertaking to rewrite mainframe tools to address the most pressing shortcomings. Another problem is that execution of office tools on these machines impacts their utility as research tools. Improvements in processor speed, memory size, display size, and window systems may address this problem in the near future. We look forward to the introduction and testing of the TI *Explorer II* and Xerox *Tamarind* with this in mind.

Some community members tried increasing their usage of Macintosh applications as a means of reducing dependence on the 2060, but except for some drawing tools, they were not up to the job, hampered by the small display and incapacity to cope with large or complex documents. We look forward to investigating the much more powerful Macintosh II as an office system and possibly Lisp delivery vehicle. Early indications suggest limited potential for Lisp development, but perhaps mass availability will encourage improvement in this area.

In the long term, we may hope to see an integration of both the Lisp machine and stock machine worlds. Despite the inadequacy of the present single-vendor offerings, the potential leverage of Lisp machine technology for office systems ancillary to research makes the pursuit of combining the two as attractive as ever, and we intend to take advantage of new hardware opportunities as they arise.

1 - Purchases This Past Year

The core resource hardware continues to be stable and the relatively small amount of SUMEX-AIM money for new purchases has been concentrated on experimental workstations and server equipment needed for distributed system development. These purchases are paced carefully with the developments of higher performing, more compact, and lower cost systems. The purchases this past year are summarized below. It should be noted that these purchases in many cases complement hardware acquired with non-NIH funding, including 3 SUN 3/75 workstations, a SUN 3/180 file server, and numerous laser printer upgrades.

1. SUN X-501B 75 Megabyte Disk Drives (3 each, for Lisp workstations)
2. Sun 6250 BPI Tape Drive (for file server backup)
3. Parity 24-Megabyte memory boards (3 each, for Lisp Workstations)
4. Apple Macintosh SE computer (for text processing and graphics)
5. Apple Mac II computer (40 Megabyte disk, 7 MB memory upgrade, and video card/monitor; for a Lisp workstation)
6. Imagen 3320-3 laser printer (for higher volume printing)
7. Ricoh 4120 laser printer (used, for spare parts)
8. Toshiba T1100 Plus Portable Computer (as a portable travel computer)
9. Ethernet (10MB bits) Multibus Interface Boards (4 each, for network expansions)
10. U.S. Robotics 9600 baud modems (2 each, for higher speed serial line connections)

1.1 - Workstation Hardware

Using non-DRR funding, the KSL has taken delivery of 20 new Xerox 1186 LISP workstations, and has upgraded 4 Xerox 1108 machines to 1109 (Dandetigers) with memory expansion and floating point support. The machines are used by many projects in the KSL, including the GUIDON and NEOMYCIN efforts, BB1, PROTEAN, and Financial Resources Management (FRM). These machines increase our research capabilities and complement the Texas Instrument Explorers and Symbolics 36XX facilities of the KSL.

Our Xerox workstations proved to be very reliable again this year and justified our strategy of saving money by not purchasing service contracts. Also to save money, we arranged with third parties to repair/replace some components that did fail. (Exception: We purchased a third-party service contract on one Xerox 1132 disk drive since the particular device has failed more than once.)

The basic components of the three Sun 3/75 workstations were purchased with DARPA funding for evaluation as AI development engines and/or office systems. Although Sun recommends these machines as general purpose workstations, experience indicated that memory and disk upgrades to the basic systems are necessary to consider their use as Lisp engines. These upgrades are on-order and evaluation is still in the early stages.

1.2 - File Server Hardware

Because our Lisp workstations have only limited local file space, the development of effective shared file servers is essential to our resource operation. SUMEX now has three UNIX-based file servers. Two of them, as reported in the past, use VAX/750's as the processors: the SAFE has four, 470 Megabyte, Fujitsu Eagle disk drives and the ARDVAX has one such disk drive. The SAFE also is equipped with a 300 megabyte CDC, removable media, disk drive and a 800/1600 BPI Kennedy tape drive. The CDC unit is used for incremental backup dumps and the tape drive is used for both incremental and full backup dumps. A procedure has been established whereby the ARDVAX is able to use this equipment for its incremental and full dumps over the network. The configurations of these systems are shown in Figure 7.

With DARPA funding this past year we bought a system called the KNIFE, a file server based on a SUN 3/180 processor. It is equipped with two of the 470 Megabyte Fujitsu disk drives and a cartridge tape drive (see Figure 5). We are in the process of adding a Fujitsu 1600/6250 BPI tape drive for backup dumping. Being relatively new, the performance of this equipment in an operational environment has not yet been thoroughly checked out at SUMEX.

The Xerox XNS Ethernet-based file server (donated by Xerox in 1985) has increased in capacity and usage in the past year. This server is based on the Xerox 8000 processor (identical hardware to the Xerox 1108 Lisp workstation but running more conventional microcode) and the Century Data Systems T-305 removable media disk drive. With the addition of two additional disk drives (also donated), the total potential storage capacity of the server has increased to approximately 900 MB (of which 600 MB is currently available from the network).

The user base for this server has grown to over sixty regular, registered users and numerous infrequent guest and project users. This server is the primary system software resource for over fifty Lisp workstations. In the past year, the server software has been upgraded twice, the most recent upgrade introduced random access to the content of files which, when interfaced to Interlisp's paged file mechanisms, should improve both the flexibility and effective speed of the server.

Though optical disks have been slow in realizing their earlier-announced potential, suitably packaged products are now appearing in the marketplace. It is possible that this technology used in place of (or in conjunction with) conventional magnetic tapes might provide an excellent medium for implementing a responsive offline storage system for data. It is fair to expect that even a small laboratory could have reasonable access to hundreds of gigabytes of storage.

1.3 - Printer Hardware

Over the past year, we purchased 2 new Imagen 12/300's, upgraded an 8/300 to a 12/300, and converted an old Hewlett-Packard 2688A to a 12/300 laser printer for the SUMEX-AIM community. These enhancements were funded by DARPA. The move to 12/300's was motivated primarily by the ruggedness of the Ricoh LP-4120 print engine used in those printers. Whereas the Canon LBP-CX print engine used in the 8/300 has an expected lifetime of 70,000 pages, the Ricoh LP-4120 has an expected lifetime of 700,000 pages. Other beneficial side-effects of the upgrade were: (1) higher print rate (12 pages-per-minute), (2) bigger paper tray (half a ream), (3) blacker and more solid print, (4) crisper print, and (5) cheaper supplies (half the price per page compared to the 8/300).

We have also acquired an Apple Laser Writer which interprets the PostScript page description language. Within a few months of its introduction, the Apple Laser Writer has become the most common laser printer on campus and around the world. Economies of scale have made it possible for us to acquire this printer for under $4000. SUMEX AppleNet/Ethernet expertise will make it possible for us to attach the Laser Writer to the high-bandwidth campus internet and operate the printer at the high-end of its 8 page-per-minute capacity. (The vast majority of laboratory-owned Laser Writers in the U.S. are driven over a low-bandwidth RS-232 line yielding only 3 pages-per-minute throughput and typically greater latency.) The PostScript page description language is already the standard of choice at university and DARPA sites (judging by traffic on the Laser-Lovers discussion group). It is generally agreed upon in these communities that PostScript is among the easiest-to-generate and most expressive of the page description languages in use today and reconciles these traits much more effectively than other languages do.

At present, most of our printers image at 300 dots per inch (dpi) and our finest printer is the aging Xerox Alto-Raven which images at 384 dpi. To exploit the special capabilities of much higher quality, camera-ready printers and to take advantage of the economical Apple Laser Writer, we have begun an Interlisp implementation of an "image stream" driver for PostScript. UNILOGIC has already added Postscript support to Scribe and Adobe has implemented Postscript support for TeX.

1.4 - Network Hardware

As we evolved a more complex network topology and decided to compartmentalize the overall Stanford internet to avoid electrical interactions during development and to facilitate different administrative conventions for the use of the various networks, we developed gateways to couple subnetworks together using Motorola MC-68000 systems. Given the heterogeneity of our environment, these gateways continually need to provide additional services to support the influx of new workstations. To accommodate current and anticipated gateway software growth, we have increased the memory capacity of the MC-68000 cpu board from 256 kilobytes to 1 megabyte.

We also developed a MC-68000 terminal interface processor (TIP) to provide terminal access to network hosts and facilities. It is basically a machine that has a number of terminal lines and a network interface and software to manage the establishment of connections for each line and the flow of characters between the terminal and host. In the past, 32 lines per TIP was sufficient, but our transition plan for moving users off the 2060 includes moving both the dial-in and dial-out functionality of the 2060 to TIPs, and this year we upgraded one of our TIPs to support 10 such ports. Thus, the 32 line upper bound is no longer feasible, and there is now the need to configure TIPs with at least 48, and perhaps 64 lines. As with the gateways, we have quadrupled the memory size of the TIPs' MC-68000 cpu board to 1 megabyte. This will adequately handle any future expansion of these servers. We have also improved the Dial IN/OUT service for both the 2060 and Tips for faster operation (2400 baud service maximum).

SUMEX-AIM is continuing its efforts in improving the networking environment for faster and more unified data communications. In this report period, several reconfigurations towards this endeavor have been completed. The SUMEX-AIM facility has been relocated to a new building. This move necessitated the relocation of all offices as well as all associated computer equipment. A network in the new building had to be designed and implemented and coupled into the old one which connects with the remaining KSL groups as well as the Stanford campus proper. This modification gave us the opportunity to upgrade several portions of the network in a manner that will provide redundancy as well as future expansion capabilities to the Medical Center and all other planned adjacent buildings. The new facility was wired to provide every sitting space with a flexible network connect capability similar to a telephone type connection. The entire scheme was successfully implemented with very little downtime. After almost a year in operation this scheme seems to be very reliable.

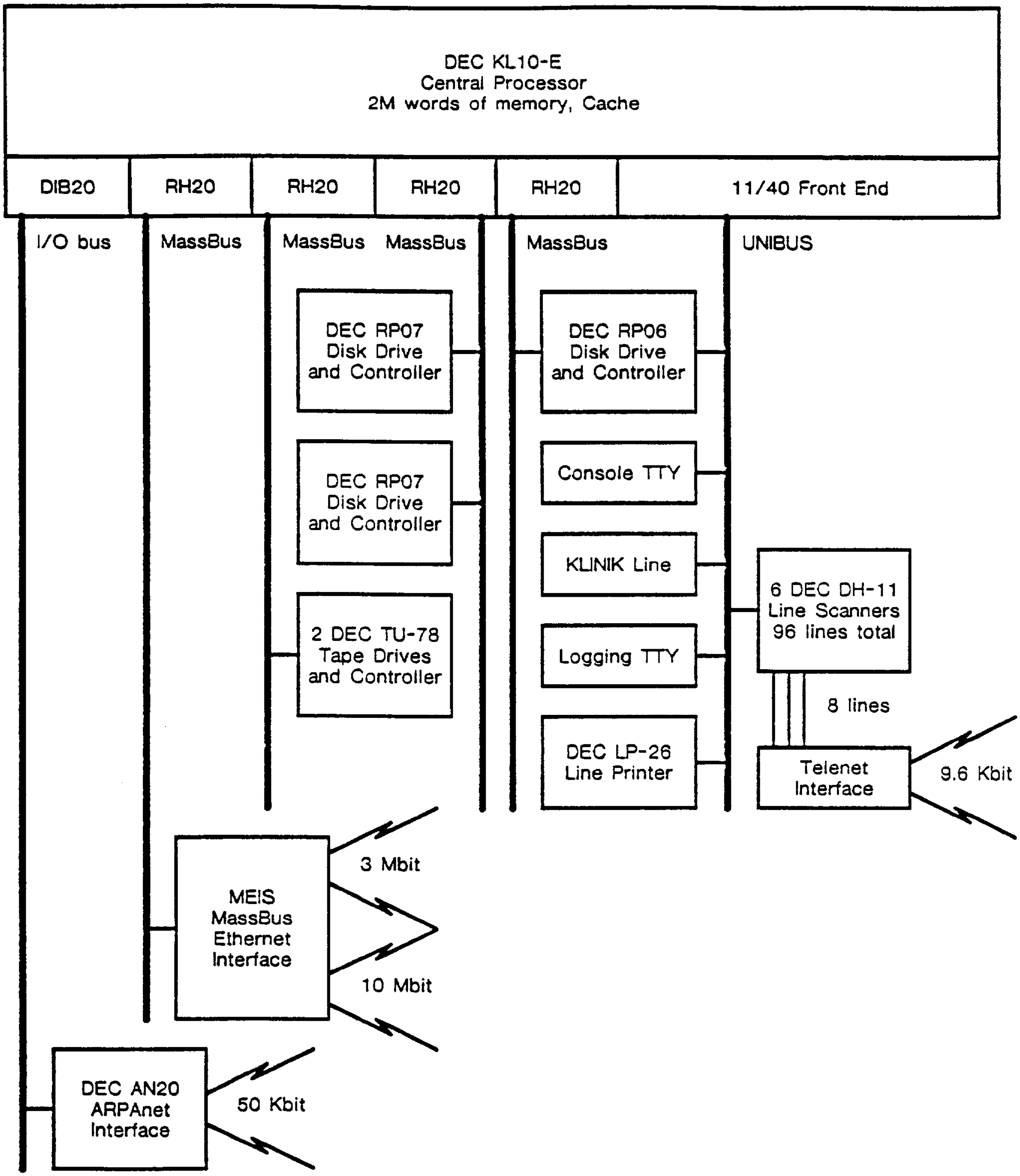

Figure 4: SUMEX-AIM DEC 2060 Configuration

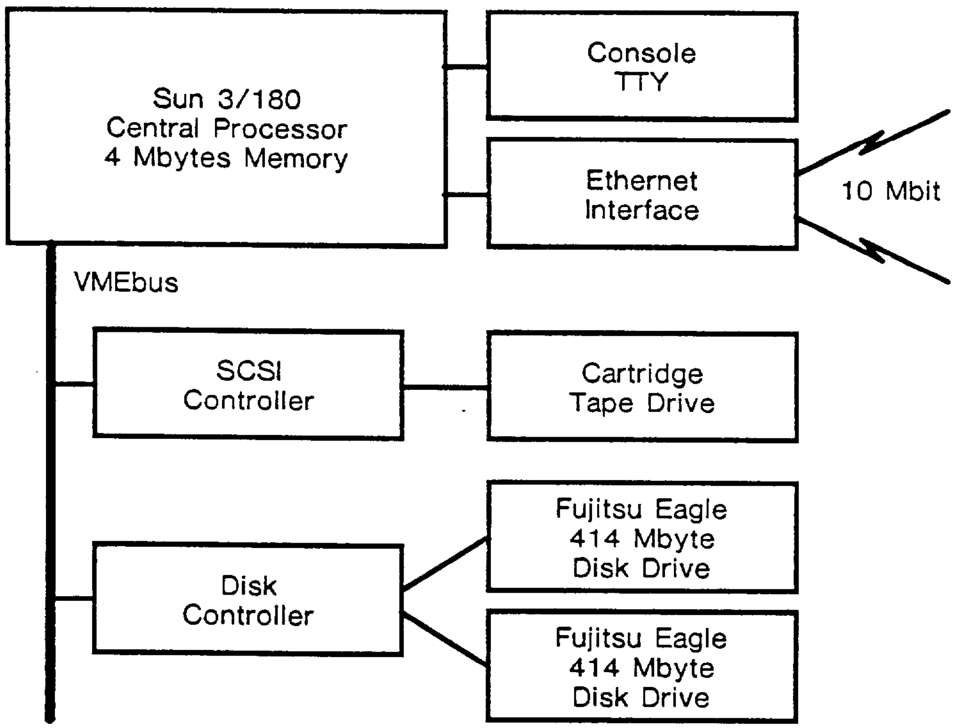

Figure 5: SUMEX-AIM Sun File Server Configuration

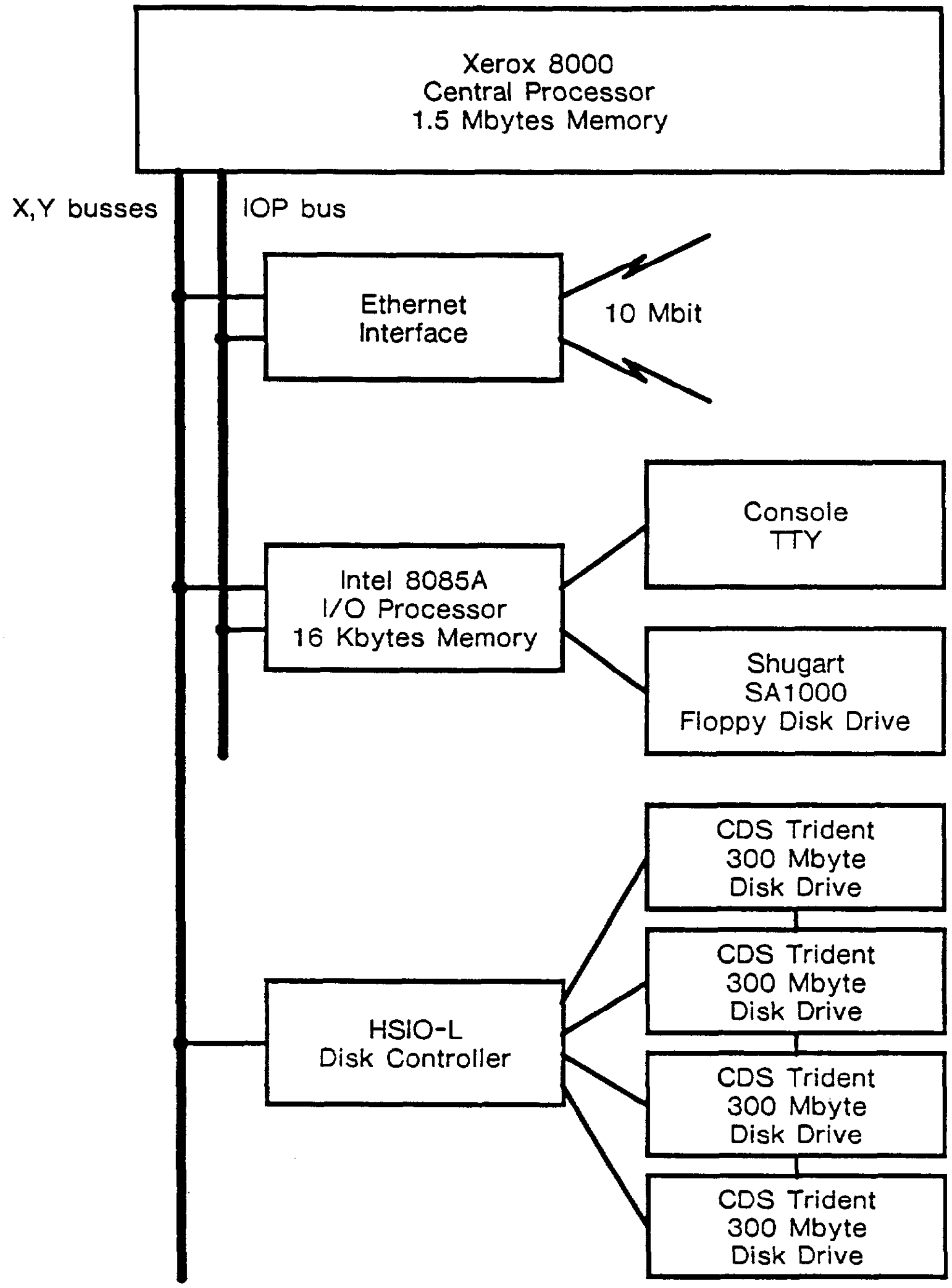

Figure 6: SUMEX-AIM Xerox File Server Configuration

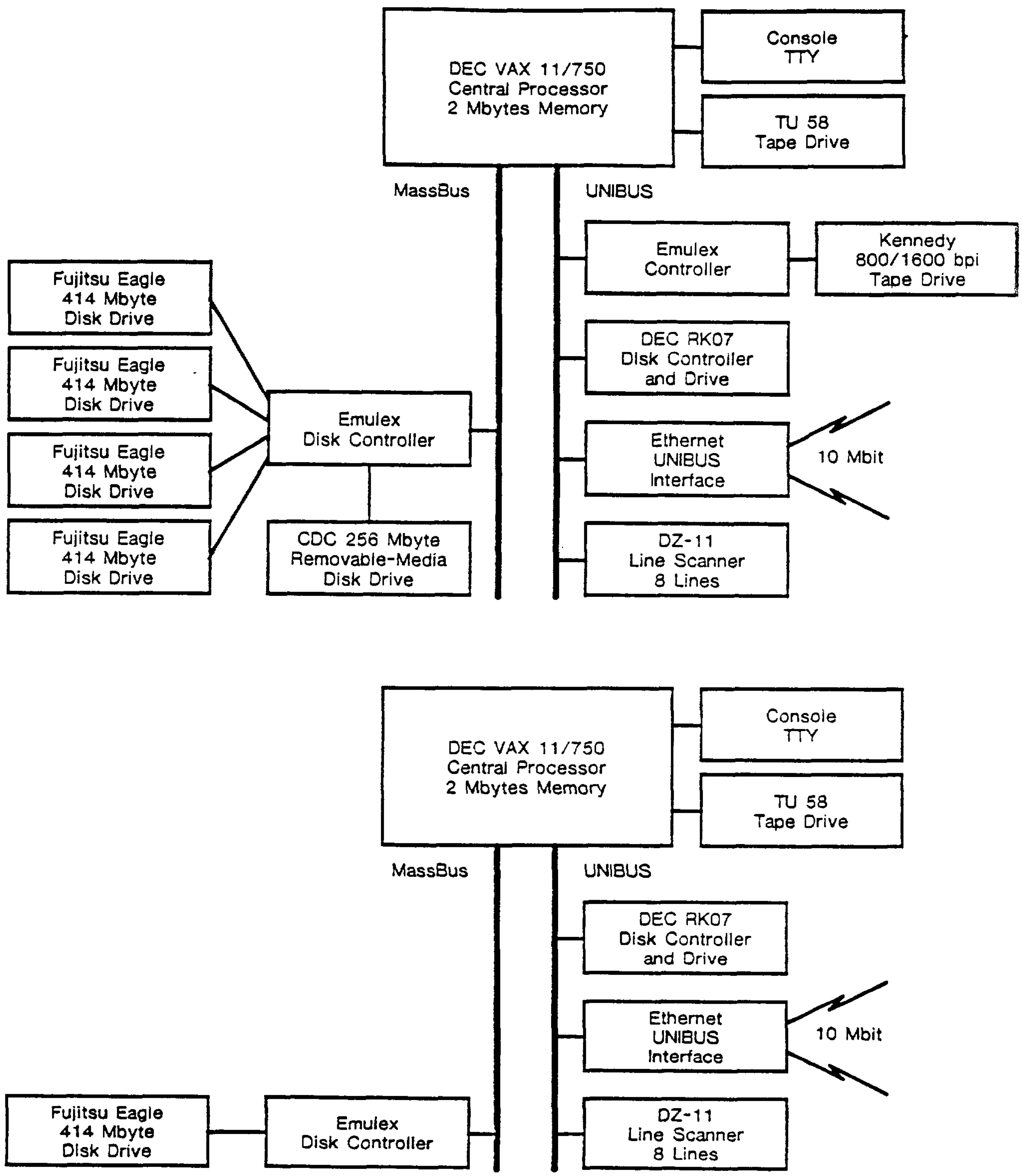

Figure 7: SUMEX-AIM VAX File Server Configuration